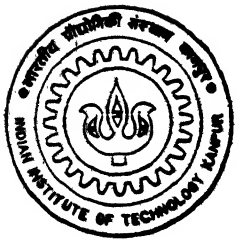
Browsing Databases on the Internet

by

SANDEEP RAO KANAPARTHI



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Technology, Kanpur

MAY, 1998

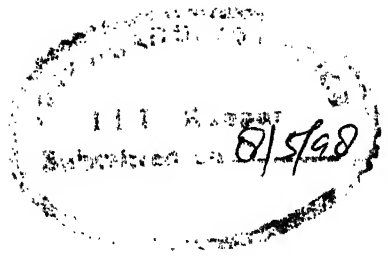
Browsing databases on the Internet

A Thesis Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Master of Technology

by
Sandeep Rao Kanaparthi

to the

Department of Computer Science & Engineering
Indian Institute of Technology, Kanpur
May, 1998



Certificate

Certified that the work contained in the thesis entitled "Browsing databases on the Internet", by K. Sandeep Rao, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

T V Prabhakar

(Dr. T. V. Prabhakar)
Professor,
Department of Computer Science & Engineering,
Indian Institute of Technology,
Kanpur.

May, 1998

Acknowledgements

I would like to express my sincere gratitude to Dr T. V. Prabhakar, whose immense knowledge in the field, with active participation and constant encouragement guided me towards the completion of the thesis. It has been great pleasure to work under his able guidance.

I would like to thank my examiners Dr Harish Karnik and Dr P. K. Kalra for their valuable comments on thesis.

Special thanks to Jeevan, Atul, Mukkam, Kataru, Dhanabal, Rajesh, Vihari and Murali for their support during the period of thesis.

Thanks to all my Hall IV friends Sairam, Srinivas, Murali for their memorable company during my stay in IITK.

Sandeep Rao

Abstract

With the emergence of Internet as the most widely used medium for information sharing, many organizations are attempting to make their databases accessible to the Internet users. Now-a-days HTML is being used for this purpose. As of now HTML does not support Visualization Techniques commonly used in datamining operations. Without visualization techniques it is not possible to present data effectively. With the emergence of Java technology, it has become possible to use visualization techniques in the browser. In this thesis we have developed a tool, Database Browser, for browsing databases on the Internet. Our objective is to design effective browsing and visualization techniques, and to minimize server and network loads.

Our tool can be divided into two parts: server-side scripts and client-side scripts. Server-side scripts retrieve data from the database and send it to the client-side scripts. The client-side scripts take input from the user, get relevant data from the server and render it in the Web browser using some metaphors.

In the beginning of a typical browsing session, the Webserver sends an initial user interface along with client-side scripts. Then onwards the client-side scripts take control and work as communicator between user and the server. These client-side scripts use some most commonly used and most generic metaphors for presenting the data in user friendly way.

Contents

Chapter 1

Introduction

The increasing importance of the Internet has spurred the creation of large number of websites. As a result, the number of users of the Internet also increased tremendously. with this many organizations are competing to bring their large databases to the potential customers/users through the Internet to accomplish their goals.

1.1 Some observations about the Internet

Here are some observations about the Internet in general:

- Users are novices. They don't know exactly what type of information is available at a particular website.
- The users don't know what is the structure of the data at the websites.
- Servers on the Internet are quiet often overloaded.
- Client machines on the Internet are almost idle.
- Internet is bursting with traffic.

1.2 Typical website goals

A primary way of categorizing websites is by goals of the originators, as interpreted by the designers. The website goals tied to typical organizations are shown in Table 1.

Purpose	Some Organizations
Sell products	Publishers.airlines, departmental stores
Advertise products	NBC, Ford, IBM, Microsoft, Sony
Inform and announce	Universities, museums, cities
Provide access	Libraries,news papers, scientific organizations
Offer services	Governments, public utilities
Credit discussions	Public interest groups, magazines
Nurture communities	Political groups, professional associations

Table 1: Typical website goals

All these organizations need to present information, available in their databases, to the users, in a user friendly way.

1.3 Typical website designer goals

The goals of website designers, according to the observation made in the above section, are:

- To minimize the load on the server.
- To minimize the data to be sent to the client .ie. convey what you want to say using minimum possible data.
- To tap the resources on the client.
- To present the information in a user friendly way.

In this thesis we designed a method for browsing databases on the Internet which satisfies above criteria.

1.4 Motivation

Lot of work has been done about accessing and presentation of databases. But presentation of databases on the Internet throws very different problems. In traditional systems the users are a specific group within the organization with some expertise. They know the type of data present and the organization of data in the databases. They have been given training about how to use the interfaces to the databases.

But users on the Internet are completely different to the users within the organization. They are novices. They don't have any idea either about the data present or about the organization of data in the databases. They don't even know how to use your user interface.

So, the criteria for interface design for databases, to present on the Internet, are completely different to the criteria for interface design for databases, to present within the organization. The users on the Internet need well known and very simple interfaces. You have to use very generic and well-known visualization techniques and metaphors for your presentation system.

Now a days people are using World Wide Web to present databases. They are presenting information using HTML documents. Using mere HTML documents is not enough to present information effectively. One cannot use visualization techniques like bar-chats and pie charts in HTML. One cannot use hierarchical metaphors like Inline expansions and Trees in HTML. So one has to go beyond HTML for presenting information effectively. Few years back a new technology called "Java Applets" has emerged. Using Applets the server can send an piece of executable code to the client machine and run the the code on the client machine. This approach suits well, to use visualization techniques on the client machine and also decreases load on the server. In this thesis we exploited this new technology for browsing databases on the Internet.

1.5 Organization of report

- **Chapter2** discusses the background issues and basic terminology of the Internet, visualization and databases.

- **Chapter3** discusses similar work that has been done in this field.
- **Chapter4** discusses our method and explains facilities provided in our method.
- **chapter5** discusses design and implementation of our method.
- **chapter6** concludes the report and suggests further extensions that can be made to the present system.

Chapter 2

Background

This chapter discusses background issues involved and required terminology.

2.1 Internet

A computer network is a collection of autonomous computers with interconnections between them. Worldwide networks gather information about several kinds of subjects. Data and results collected in several experiments in different parts of the world are preserved and are used in similar experiments elsewhere which boosts the growth of research and development. Supercomputers can be shared to utilize the computing power to their maximum extent possible. Several other kinds of resources like printers, disks, etc., are also shared. All these are the results of computer Networks.

But it is not possible to build a single universal network out of a single hardware technology. Some groups of users need high speed networks which are not possible over long distances and some groups need communications for long distances which are somewhat slower. So, different users choose different hardware technologies which meet their communication needs. Because of these reasons, for some time in history, different networks existed in independent entities. The new technology that evolved a few years back allows multiple, diverse networks with different hardware technologies by adding both physical connections and a new set of conventions. This is called

internetworking or internetting. Internetting work over different hardware technologies and permits computers to communicate independent of their physical network connections. The large network of computers formed by internetting is called an Internet.

Internet is used to describe the massive world-wide network of computers but it literally means network of networks. The Internet comprises of thousands of thousands of smaller regional networks scattered throughout the globe. There is no single governing body that owns the Internet[9]. Each network in the Internet is owned and controlled locally with its own local policies.

Internet provides basic services such as electronic mail, access to information resources, network news, and ability to transfer files. Having access to Internet usually means having access to these services[11].

2.2 Internet applications

The net is merely a platform. The applications make it happen. Here is the list of basic Internet applications:

- **email** - Correspond with people, or even groups of people.
- **Usenet news** - Read and post public messages in newsgroups covering thousands of topics.
- **file transfer (FTP)** - Retrieve programs, documents, and other files.
- **gopher** - A menu-based system for retrieving documents and information.
- **The World-Wide Web** - A flexible system for reading hypermedia and multimedia.

The Table 2 below elaborates the purpose of Internet Applications.

The WWW's ability to provide interactive communication and to link documents makes it the ideal application for browsing databases on the Internet.

	Interactive Communication	Research	Resource Acquisition	Fun
email	yes	-	-	yes
Usenet news	yes	yes	-	yes
FTP	-	-	yes	yes
gopher	-	yes	yes	yes
WWW	yes	yes	yes	yes

Table 2: purpose of Internet applications

2.3 WWW

WWW stands for "World Wide Web". The WWW project, started by Tim Berners-Lee while at CERN (the European Laboratory for Particle Physics), seeks to build a "distributed hypermedia system." In practice, the web is a vast collection of interconnected documents, spanning the world [15].

The advantage of hypertext is that in a hypertext document, if you want more information about a particular subject mentioned, you can usually "just click on it" to read further detail. In fact, documents can be and often are linked to other documents by completely different authors - much like foot-noting, but you can get the referenced document instantly! To access the web, you run a browser program. The browser reads documents, and can fetch documents from other sources. Information providers set up hypermedia servers from which browsers can get documents.

The browsers can, in addition, access files by FTP, NNTP (the Internet news protocol), gopher and an ever-increasing range of other methods. On top of these, if the server has search capabilities, the browsers will permit searching of documents and databases. The documents that the browsers display are hypertext documents. Hypertext is text with pointers to other text. The browsers let you deal with the pointers in a transparent way – select the pointer, and you are presented with the text that is pointed to.

Hypermedia is a superset of hypertext – it is any medium with pointers to other media. This means that browsers might not display a text file, but might display images or sound or animations.

2.3.1 Hypertext and Hypermedia

Hypertext is text with links. Hypertext is not a new idea: in fact, when you read a book there are links between references (eg. "see section 3"), footnotes, and between the table of contents, or index and the text. If you include bibliographies which refer to other books and papers, text is in fact already full of references.

With hypertext, the computer makes following such references as easy as turning the page. This means that the reader can escape from the sequential organization of the pages to follow pursue a thread of his or her own. This makes hypertext an incredibly powerful tool for learning. Hypertext authors design their material to make it open to active exploration, and in doing so communicate their information and ideas more effectively.

WWW uses hypertext as the method of presentation, although as we shall see, this does not necessarily require that authors write hypertext. In WWW, links can lead from all or part of a document to all or part of another document. Documents need not be text: they can be graphics, movies and sound, so the term "hypermedia", meaning "multimedia hypertext" applied equally well to WWW [7].

2.3.2 HTML

The standard language the Web uses for creating and recognizing hypermedia documents is the Hypertext Markup Language (HTML). It is a subset of the Standard Generalized Markup Language (SGML), a method of representing document formatting languages. Languages such as HTML which follow the SGML format allow document writers to separate information from document presentation - that is, documents containing the same information can be presented in a number of different ways. Users have the option of controlling visual elements such as fonts, font size and paragraph spacing without changing the original information.

2.3.3 HTTP

Web software is designed around a distributed client-server architecture. A Web client is a program which can send request for a document to a Web server. The Web server is another program, which upon receiving the request from the Web client, sends the document requested. The task of storing the document is left to the server and the task of presenting the document is left to the client. Each program concentrates on its own duty and progresses independent of each other. This is purely distributed in nature. The language the Web servers and the clients use to communicate with each other is called Hyper Text Transfer Protocol(HTTP). The Web servers are often called HTTP servers. World Wide Web is also used to refer to the collective network of servers speaking HTTP.

HTTP is an application level protocol. It is generic, stateless and object oriented protocol[15]. It can also be used for other distributed object management systems through extension of its request methods. The typing and negotiation of data representation is an important feature of HTTP.

HTTP allows two types of documents to be requested by clients.

- **Static documents:** The first type of requests are for simple files stored on the server. The files stored on the server will be accessed directly and served. This type of documents are called static documents. The static documents are used to present information that doesn't change or the information in which changes cab be done manually. They may also contain links to other documents. Static documents are created easily by storing normal text or hypertext in a disk file. The documents cab be delivered quickly just by accessing the disk without executing a program.

- **Dynamic documents:** The other type of requests are for a document which will be generated by the server on the fly. Such documents are called dynamic documents. The dynamic documents may never exist on disk at all. In some dynamic documents, there will be large portions of fixed content with a small amount of dynamic content generated when the page is actually delivered. The dynamic content can be data from a database or from a scientific

instrument. The server-side include mechanism is one approach to dynamic documents. Even though the current form of HTML doesn't allow this, Web servers can provide their own version of HTML in which some tag will be used to include another HTML file. When a document is requested, the server will expand all the include files and serves the document as one single document. The dynamism increases when the include document is the output from a program or another dynamic document. The limitation here is the performance of the server. CGI is the most commonly used approach for generating dynamic documents. In CGI, documents are generated completely outside the server by executing an external program. The input is given to the program by the server and the output of the program is expected from the standard output of the program.

2.4 Accessing a database server via the World Wide Web

2.4.1 Why use the World Wide Web?

- **Graphical User Interface:**

  One of the main issues in using a database is the problem of accessing the data. Many databases provide a programming interface for computer languages such as C or Fortran, or a text-based menu-driven interface such as SQL, but either of these can be cumbersome and/or confusing to use. A Graphical User Interface (GUI) can make database access easier, but the implementation of such an interface can require programming expertise specific to a hardware platform, and most database users should not be required to know how to implement a GUI [13].

  Access to the World Wide Web is provided by a Web browser. Web browsers provide a GUI that can be used to access many things, including a database. Using a Web browser's built-in forms capability, users can access a database by simply filling in the data they want and pressing a button. The returned data

can then be presented in an easily readable format.

- **Standardization:**

  Web browsers access documents on machines around the world. These documents are composed of HyperText Markup Language (HTML) formatting instructions and text, as well as graphics and links to other Web documents. HTML is a standard to which all Web browsers adhere. An HTML document on one machine can be read by users on any machine in the world, provided they have a Web browser and a connection to the Internet. By using HTML as a document standard, programmers only learn a single language, and users only learn to use a single GUI. By using a standard set of tools, code maintenance is simplified and new ideas can be implemented more quickly. Also, Web browsers provide a utility to view the HTML source document of any Web page. This makes it very easy to find out how to construct HTML documents by simply finding a page that does what you want, and viewing the source code.

- **Cross-Platform Support:**

  Web browsers are available for virtually every type of computer. While operating systems and GUI's exist in bewildering numbers, there is no need to learn them all, or to limit an application to just a handful. Web browsers and HTML provide a means to access the World Wide Web from many different types of machines.

  This cross-platform support allows users on most types of machine to access a database from anywhere in the world. Information can be disseminated with a minimum of time and effort, without having to solve compatibility problems.

- **Network Access:**

  One of the benefits of having an HTTP (HyperText Transfer Protocol) server, which allows users to access HTML documents on your machine is, the networking overhead is handled by the HTTP client and server and neither the user nor the HTML programmer need to know anything about the network or how to use it.

This built in network support greatly simplifies the task of database access by eliminating the problem of getting two different hardware platforms to talk to one another, as well as eliminating the expense of buying additional networking software.

2.5 Method of Database access

The common method of accessing databases through WWW is by using CGI scripts [6]. HTTP protocol allows external programs to be run in response to particular requests from the client. These external programs are called CGI programs and can be used to access the database. CGI programming is discussed in detail in later sections. Various interfaces involved in accessing a database are shown in Figure 1. The scheme used in the figure is a generic construction. Specific implementations may take any number of variations on this scheme or may chose to bypass it completely. In this scheme, processing software is the actual CGI program. Interface software is database-specific interface used to translate the query into a format recognized by the database and is used by the processing software. Accessing software is the actual programming API or command-line interface distributed along with the database. It takes formatted query from Interface software(IS) and sends the results to Processing software(PS) via Interface software(IS).

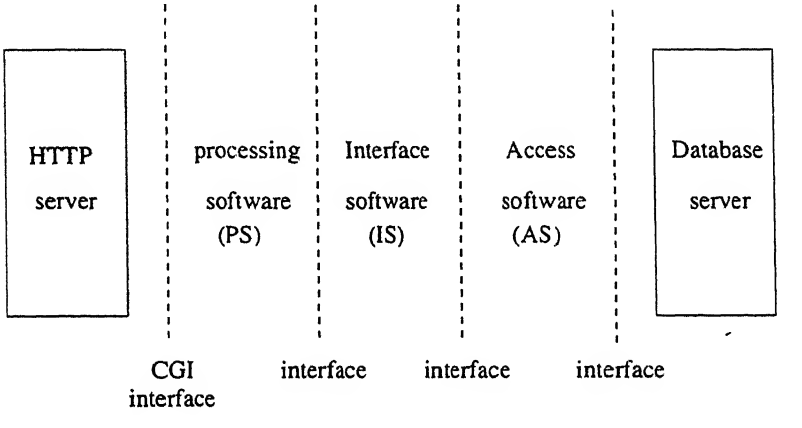


Figure 1: method of database access

2.6 CGI

Web servers are able to deliver static documents from files without assistance of other programs. However dynamic contents, database interfaces and the ability to accept user input cannot be accomplished by static documents alone.

2.6.1 Need for CGI

In the beginning, Web programmers knew how to generate Web pages on the fly. There were servers developed solely to deliver a particular type of dynamic document. The same technique is sometimes followed when efficiency is of primary concern. Clearly, this is not practical, because the advanced security, logging, communications and load-management features of Web servers cannot be replicated for every application that require dynamic documents. So, Web server developers began to provide interfaces through which external programs could be executed, in response to a request for a document in a designated portion of the document tree. Different types of Web servers used to have their own specific interfaces. As a result , it was impossible to write a single external program that would be executed with all servers. Thus a strong need for the standardization of external program interface was felt.

The NCSA(National Center for Supercomputing Applications) and the CERN(European Laboratory for Partical Physics) developed a standard interface which is now known as Common Gateway Interface(CGI). Common Gateway Interface(CGI) is a standard interface for the construction of completely dynamic documents generated by programs installed externally on the server system. CGI is intended to provide a consistent interface between Web servers and programs that extend their capabilities. CGI extends the capability of World Wide Web by allowing a program to generate Web documents on the fly even by accepting user input. The CGI program can use any resources available to the Web server to accomplish its task.

2.6.2 Goals of CGI

The following are the goals of CGI:

- **Consistency:** The main goal of CGI is to provide consistent interface between the Web server and the external program.
- **User input deliver guarantee:** In addition to a consistent interface, CGI standard seeks to provide a reasonable guarantee that user input particularly from form submissions, will not be lost due to the limitations of the server operating system.
- **Additional information:** The CGI standard also attempts to provide the external program with as much information as possible about the server and the browser, in addition to any information that may be known about the user.
- **Straight forwardness:** The CGI standard attempts to be straight forward as to make the development of simple CGI applications easy.

The processing of form data in CGI programming is somewhat complex. Even then CGI programming is made easy by the freely available tools including several several libraries in commonly used CGI languages such as Perl,C and PL/SQL. The portability and simplicity of CGI standard has led to its widespread use.

2.7 Visualization

Visualization is a process of presenting information using visual components such as graphs, charts, glyphs etc,.

As somebody said, "A picture is worth thousand words"; human beings can perceive lot of information about similarities and differences in patterns, effectively and instantaneously by looking at a picture.

2.7.1 Purpose of data visualization

- Harness perceptual capabilities of human visual system to extract information from data sets [8].
- Look for structure, features, patterns, trends, anomalies, relationships.

- Provide a qualitative overview of large, complex data sets.
- Assist in identifying region(s) of interest and appropriate parameters for more focussed quantitative analysis.

2.7.2 Methods of data visualization

Here are some simple and most commonly used data visualization techniques.

Pie charts

A pie chart, shown in Figure 2, is a circle divided into component parts. The proportion of the whole that each part contributes is represented by the relative size of the slices. These charts display simple proportions within a single group and describe how the component parts contribute to the whole.

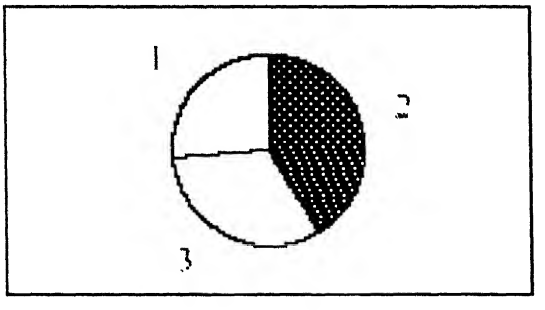


Figure 2: Pie charts are best for showing the relative proportions of parts of a whole

Pie charts should be used carefully, because they do not show the actual values measured, but show only the proportion of a whole. Therefore, use a pie chart to describe the parts of a group, not the size of the group itself. Note, too, that comparisons generally cannot be made between adjacent pie charts. Use bar charts to compare proportions among groups.

A pie chart can only be used when the data represents portions of a group total, such as income by category or population by age group. Do not use a pie chart if your data does not add up to the total, if you do not have data on all parts of the

group, if the parts overlap one another, or if you are trying to describe more than one group.

Pie charts should only be used when there are relatively fewer component parts (generally six or fewer). Too many parts tend to clutter a pie chart. When there are large numbers of component parts, use a bar chart instead.

Bar charts

A bar chart, shown in Figure 3, is a series of bars extending from a baseline. The relative size of each group is represented by the length of each bar. Bar charts are used to describe the relative magnitude of a measure across discrete groups or to show direct comparisons between parts, groups, or categories. These charts highlight contrasts among groups.

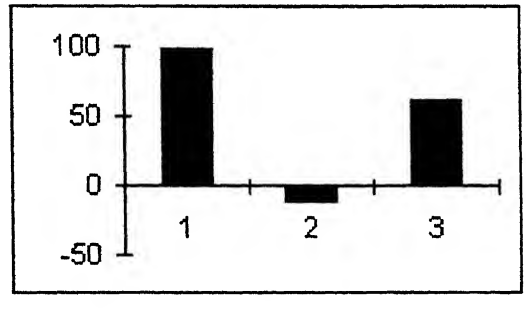


Figure 3: Bar charts or column charts are best to show comparisions of data across time or among groups

Line charts

A line chart, shown in Figure 4, is created by connecting a series of data points with a line. The magnitude or frequency of a measure is shown along the Y axis. Time is generally along the X axis. Line charts demonstrate changes in data over time or along some other continuous measure. Comparisons between groups are shown by plotting each group on a separate line. This form of graph emphasizes the trend, but is not effective at expressing actual data values.

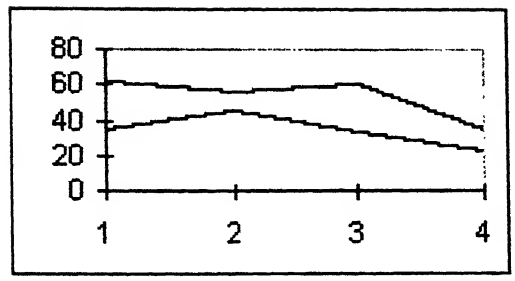


Figure 4: Line charts demonstrate changes in data over time or some other continuous measure

Line graphs are used only when the data represent a continuous measure, such as time or weight. Continuous measures are those that represent an uninterrupted progression from one value to the next. Time and date are continuous variables. Project number, department, and population group are not continuous measures. If you are not sure, do not use a line graph; use a bar chart instead.

2.7.3 How to Use Graphs to Present Data

- **Step 1: Know your purpose**

  The first step in preparing effective graphics is to have a clear purpose. Think about the information you intend to convey with the graph. Are you exploring summary tabulations, reporting on specific questions, or demonstrating a finding? If you are exploring data, a rough graph can help you quickly spot trends and identify outlines. But if you are presenting results or making a specific point, select only the pertinent data to keep your message clear. Superfluous data will cloud your message, so make sure you focus on your purpose.

- **Step 2: Know your audience**

  Next, consider your audience. Will the data be presented to researchers who will understand group comparisons or to a general audience that may require simplified comparisons? Will the material be presented in a printed report, at a meeting, or at a large convention? Larger audiences generally require larger type sizes and fewer words, so the graphs can be seen across a large room.

- **Step 3: Select your graph type**

  With your purpose and audience in mind you can begin to develop your presentation material. Be sure to use the correct graph type for your data. The most suitable type of graph depends on the type of data being displayed. Different graph types imply different concepts and serve different purposes.

  Pie charts show simple proportions within a single group.

  Bar charts (either vertical or horizontal) indicate magnitude across discrete groups or show comparisons between parts, groups, or categories.

  Line charts show changes in data over time or over repeated measurements of variables under varying conditions.

2.7.4 Designing Effective Visualizations

- Key (legend) and labelled axes essential to interpretation
- Use "intuitive" mappings where possible (spatial to spatial), though sometimes the non-intuitive mappings can reveal interesting features
- Use color with care - be aware of context-sensitive color expectations, provide ready access to alternate color maps, user customization
- Provide easy methods for view selection and modification
- Avoid overcrowding images - provide users with opportunities to enable or disable features
- Avoid distorting data (viz lies)
- Scale your data with care, and convey scaling in key
- Don't compare apples and oranges - e.g. the correlation between sun spots and the stock market
- Be concise - avoid excessive gimmicry (e.g. 3-D graphics for 1-D data)
- Avoid dependence on absolute judgments - relative is more reliable

- Differentiate original data from derived (e.g. interpolated, smoothed) data
- Don't forget aesthetics (visualizations should be appealing to the eye)

2.7.5 Metaphors

Metaphor, as the term is used in interface design, has only a little in common with metaphor in literature. Metaphor in literature is "an implied comparision between two things of unlike nature that yet have something in common"[10]. Both things are known, and the power of metaphor is in an unlikely or surprising pairing. With the pairing, the author tells us something surprising about what we thought we already knew. In a computer interface, one of those things is new and unknown. The power of the metaphor is in making a new system look and act like a known system. Interface metaphors provide the user with a user model directly.

For example, an abstract function may need to be invoked by having the user click the mouse pointer in a particular location. The clickable spot can be indicated by a group of coloured pixels or by a highly rendered image of a lifelike button. In the latter case, we use a metaphor of a physical button, which has a function and appearance that users are familiar with already; when we apply a metaphor we apply its function and appearance to the screen.

Linking a model of a known system and its functions to an unknown program via metaphor allows the user to apply what they know about the one system to the new one. This link applies the user model of the known system to that of the unknown system. Users then make certain assumptions about the new system. For example, they can assume that Macintosh Finder folders can be opened and that objects(files and other folders) can be placed inside of them or removed from them.

2.7.6 Metaphors for hierarchical structures

Hierarchical structures are the most commonly used and most generic structures for browsing databases. Here we will discuss a couple of metaphors used for presenting hierarchy structures.

Inline expansion is the most famous metaphor used to present hierarchy structures, on the desktop. Figure 5 shows an example of Inline expansion, used in windows environment.

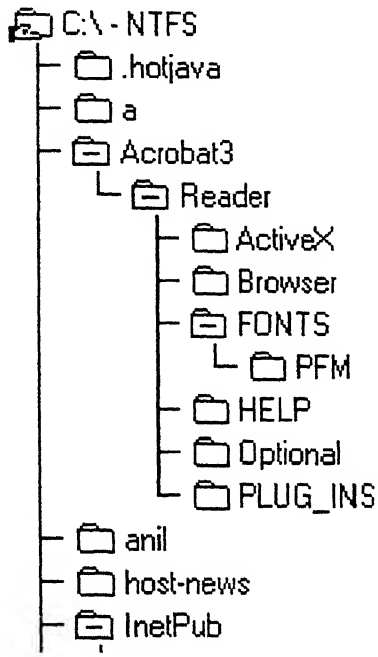


Figure 5: an example of Inline expansion

Tree structure is used on Sun-OS(Sun Operation System), for showing directory structure. Figure 6 shows an example of tree structure.

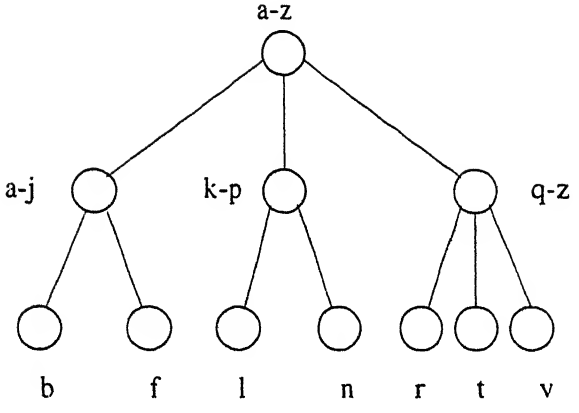


Figure 6: an example of Tree

Chapter 3

Related Work

3.1 Methods of accessing database

Lot of research has been done about how to access databases. It has come a long way from primitive text based SQL to recent data mining and visualization. There are broadly two methods for accessing a database, querying and browsing.

3.1.1 Querying

In this method user issues a query statement to the database access software. The most commonly used language for querying a database is SQL(Structured Query Language). This is a standard. By using SQL, one can query any database. To use SQL, on needs to be trained.

Many application programs have been developed using API(Application Programming Interface) provided along with the database server software. Some examples of these API's are Pro* C, Oraperl for Oracle. These application programs simplify the query process for the user. They take relevant information from the user such as field names and values and construct SQL statements by themselves and returns the results in the format, the user needs.

To use querying, one needs to know the structure of data. Another disadvantage is, an ill-formed query can be very costly for the database server and also for the

network.

3.1.2 Browsing

In this method, the user will be presented a broad view of the database. By clicking on any related topic, table or attribute, one can zoom into the particulars. Like that one goes on browsing the database until he finds the relevant information. In this method, the user need not express exactly what he wants. He just needs to recognize the related information.

3.1.3 Querying Vs Browsing

Three reasons why users might prefer browsing to analytical strategies:

1. Many users, especially novices are unwilling or unable to cogently formulate their search objective.

2. Browsing places less cognitive load on the user. In browsing a user only has to recognize a term related to the search objective. Where as in an analytical search strategy, the user has to recall and/or formulate the search query. The conditions for matching are thus far less stringent for browsing.

3. Some information systems support and even encourage browsing especially if there is no time penalty associated. with this form of information search. Although the purpose was to compensate for the lack of special clues, the fast response time also made it easy for users to browse. Because the time to retrieve a topic was nearly instantaneous, users could safely explore the system knowing that it would be quick and easy to return to the previous topic.

3.2 Data mining and Visualization

Data mining: The process of automatically extracting valid, useful, previously unknown, and ultimately comprehensible information from large databases and using it

to make crucial business decisions. OR "You torture the data until they confess" .

Data mining is used to retrieve information from the database and visualization is used to present the information in user friendly way.

A lot of work has been done in this area also. Lot of Data mining and visualization products are also available in the market. Data mining software works at server side and visualization software works at client side. So, the user needs this visualization software at his end. This approach cannot be used on the Internet, because the user has only a browser to browse Internet.

3.3 Databases and Internet

Visualization of databases on the Internet is very new area. The emergence of platform independent language, Java, and it's Applet technology which allows to send an executable code to the client and execute it on the client machine, made visualization of databases on the Internet possible.

We used this approach to develop a tool which allows visualization of databases on the Internet.

Chapter 4

Database Browser

In this chapter we describe the facilities, incorporated in our Database Browser, in detail. Our tool enables the user to browse any table using any of the four metaphors mentioned below. The user can browse the table on any field. The table is presented using multilevel index so that the server need not send whole index of a table to access a single record. The issues we have considered in this work broadly fall into two categories. One is networking part and the other is browsing and visualization part. The following issues have been considered in networking part.

- Minimizing the load on the server.
- Minimizing the data to be sent to the client.
- Utilizing the resources on the client as much as possible.

The following issues have been considered in the browsing part.

- Using very simple, and easy to use metaphors to present data.
- Selection of the most commonly used techniques so that the user need not waste time on learning how to use the technique.
- Selection of the most generic metaphors so that they can be used to present any data.

4.1 Facilities incorporated

The following facilities have been incorporated in our tool.

- Server-side facilities
    1. The server can make any table in the database available to the client.
    2. The server can present multiple indices on a table to the client.
- Client-side facilities
    1. User can select any table, key and metaphor combination and browse the table.
    2. User just needs to use mouse, and need not touch the keyboard.

4.2 A Typical Browsing Session

The following sequence of events are in a typical browsing session.

1. The client machine sends a request to the server to browse the database.
2. The server machine sends list of tables, list of fields on which each table can be indexed, a set of metaphors which the user can use to browse the table, and also a script which coordinates the user input and constructs the corresponding URL.
3. The user selects a table, a field, and a metaphor. After the user clicks on a table button, the script at the client-side collects all the input, and sends a request to the corresponding script on the server-side.
4. The server receives the table name, field name and metaphor name. Then it constructs tree index of the table on the field, and sends the top level index values to client along with the applet corresponding to the metaphor specified by the user.

5. From this step onwards the applet takes control from the web browser, and performs the communication and rendering jobs by itself. The user can access any subtree of the index tree by just clicking on the parent node of that subtree. The applet takes input from the mouse and brings the relevant subtree from the server and renders using metaphors. Like this, the user can browse whole index tree. If the user clicks on a leaf node then the applet brings the corresponding record and displays it in the adjacent window.

We have implemented the following four types of hierarchical metaphors to present the database.

- simple hierarchy: This uses simple HTML documents with buttons to present the database. First, this will show the highest level of hierarchy. If you click on one button, this will show the corresponding next level and so on. If you click on a leaf node, this will show the corresponding record in the adjacent window.

- Tabbed index: This uses simple HTML documents with links. This uses tri index instead of tree index. At the top level, this will show single characters A to Z. If you click character 'X', the next level index shows the strings from XA to XZ. Like that, the substring grows until the tri reaches its leaf-node. If you click on a leaf node, this will show the corresponding record in the adjacent window as in the case of simple hierarchy.

- Tree index: An example of tree is shown in Figure 6.This uses applet to show the tree. If you click on a node, that corresponding subtree expands. The tree grows until it reaches its full length. One can also collapse a subtree by just clicking again on the expanded subtree.

- Inline expansion: An example of Inline expansion is shown in Figure 5. This also uses applet to show the hierarchy structure. One can expand and collapse a node by just clicking on that node.

Some snapshots of the system are shown in Figures 7, 8, 9, 10, and 11. Each figure is having three windows.

Figure 7: An exapmle of Simple Hierarchy

Figure 8: An exapmle of Simple Hierarchy

Figure 9: An exapmle of Tree Structure

Figure 10: An exapmle of Inline Expansion

Figure 11: An exapmle of Tabbed Index

The bottom window is Input Window(IW). In the beginning of a session; the tables, fields and metaphors are shown in this window along with radio buttons and simple buttons as shown in the figures. The user selects table, field to be indexed on and metaphor from this window.

The top-left window is indeX Window(XW). This window shows the index tree. In this window the user can browse through the index tree.

The top-right window is Record Window(RW). This window shows the records in a table. If a user clicks on a leaf node of a index tree in XW, the corresponding record is shown in this Record Window.

Chapter 5

Design and Implementation

5.1 Architecture

5.1.1 Server-side architecture

As Oracle[3] is the most widely used DataBase Management System in the world and also provides a lot of tools for web programming, we have chosen it to implement our system.

The Oracle WebServer[1] is made up of four components. These components work together to accomplish the task of static and dynamic document delivery to the Web clients. The four components are:

- **Oracle Web Listener:** The Oracle Web Listener receives client requests. Requests for the static documents are handled internally by the Listener. In the case of serving static documents, the Listener functions just like an HTTP server. In case of dynamic documents, it just invokes the the Oracle Web Agent.
- **Oracle Web Agent:** The Oracle Web Agent handles requests for dynamic documents. It invokes the requested procedure and passes the input to it. And, it also transmits the output of procedures back to the Web Listener.

- **Oracle WebServer Developer's Toolkit:** This is a set of packages containing procedures used to create dynamic documents.
- **Oracle Server[2]:** This actually stores the data in the form of relational tables. The procedure contact with Oracle Server to retrieve data from the database.

Figure 12 shows the interaction between the components.

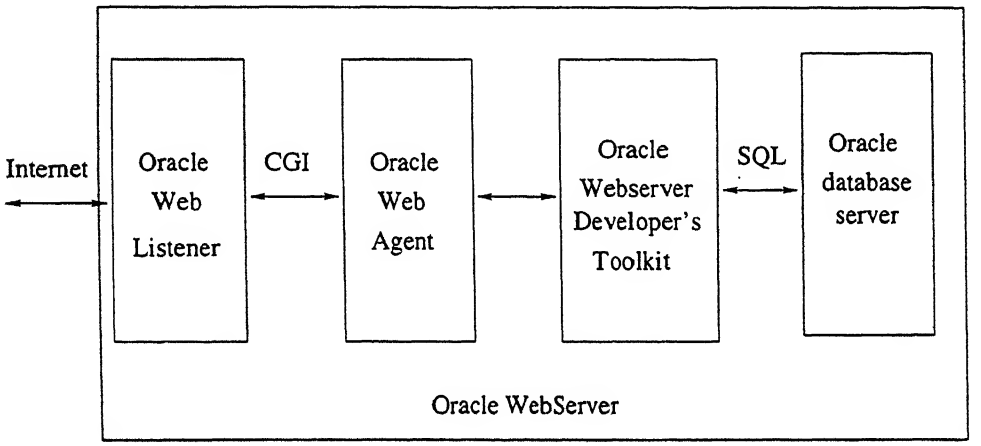


Figure 12: Server-side architecture

The server side scripts fall into Oracle Webserver's Toolkit.

5.1.2 Client-side architecture

The Figure 13 shows the client side architecture. Browser allocates some space to the Applet in the browser real estate. The Applet is allowed to render anything in this space only.

5.2 Design

Server side scripts have been developed in PL/SQL[5][4]. Three PL/SQL procedures have been written. First one serves data needed for Simple Hierarchies; second one serves data needed for Tabbed Indices and the third one serves the data needed for

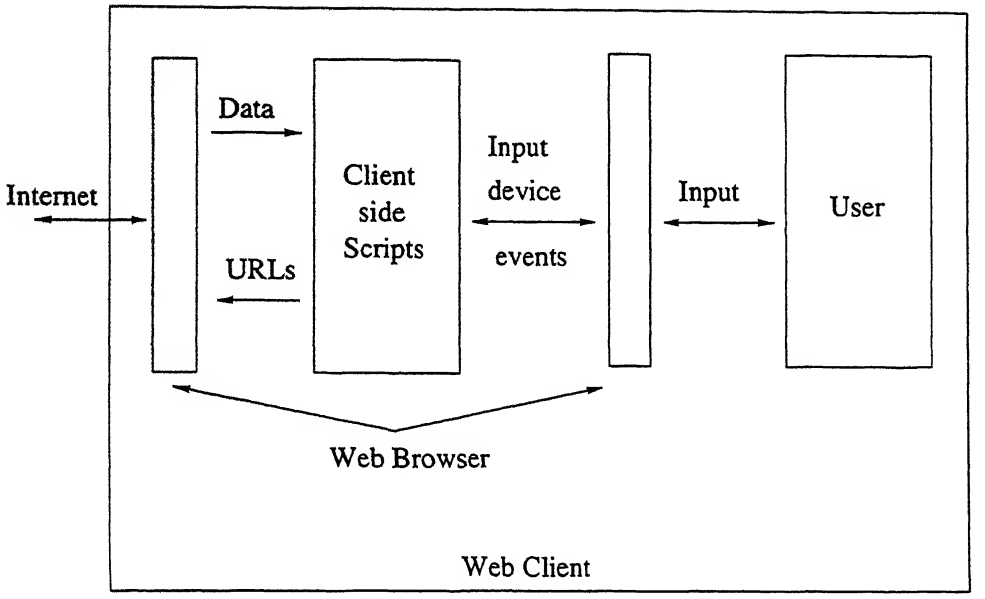


Figure 13: Client-side architecture

Inline Expansions and Tree structures. The output of the first two scripts are in HTML form. They don't need any client-side script to show the output of these scripts on the client machine. The browser look after this process. But the output of the third script is just raw data. It needs a client-side Java Applet to show this data on the client machine in a user friendly way.

Client-side scripts have been developed in Java[11] and Java Script[13]. The Java Script takes all the input from the input window and calls the corresponding server-side script. It also sends the user input such as table name, field name and metaphor name to the server-side script.

Java Applet renders the Inline expansion and Tree metaphors in the Index Window. If the user clicks on a node of the Index tree, it expands that node. If the node is already expanded then it collapses that subtree.

We have chosen PL/SQL for the server-side scripts because it is very easy to write CGI programs as well as to access Oracle server using PL/SQL. Oracle Web Agent looks after all the CGI programming issues and provides transparent interface to PL/SQL procedures. PL/SQL programmer need not know CGI issues in detail.

5.3 Implementation

5.3.1 Server-side scripts

There are three server-side PL/SQL procedures, namely showTableTab.sql, showTable-Tree.sql and Index1.sql. All these procedures takes table-name, field-name, hierarchy-level, field-lower-bound and field-upper-bound as inputs. Arity-of-the-index-tree is a constant. Suppose the arity of the index tree is X and the number of records in between lower-bound and upper-bound values of the index field is Y. Then each node in the next level of this subtree is parent of Y/X number of children. Each node in the next level of this subtree is also having it's own lower-bound and upper-bound values. The scripts then send these next level nodes to the client. showTableTab.sql corresponds to Tabbed Index; showTableTree.sql corresponds to Simple Hierarchy; and Index1.sql corresponds to Inline Expansion and Tree structure. The hierarchy parameter tells whether the node is root-node or intermediate-node or leaf-node.

5.3.2 Client-side scripts

On the client side there are one Java Script and Java Applets. The Java Script collects user input from radio buttons and simple buttons in the Bottom Window. Then it invokes the corresponding script on the server-side with suitable parameters. The Java applet is responsible for rendering the Inline Expansions and Tree structures. It takes mouse pointer coordinates as input and maps these coordinates to Index tree node. Depending upon the node values such as lower-bound and upper-bound it invokes the server-side scripts with corresponding parameters.

5.4 Installation

The steps to install the scripts are shown below. Here it is assumed that the database tables are already available.

1. Load the PL/SQL procedures and WebServer developer's Toolkit in to the account of a user, whose tables you want to present on the Web. Usually the user

account SCOTT(which is a public account in Oracle WebServer).

2. Install oracle Web-listener on a port. We have used port number 9998.

3. Create an Oracle WebServer Agent corresponding to the user on the Oracle Web Listener. We have used an OWA named demo.

4. Place the HTML files, applet.html, left.html,right.html, table.html and index.html in some directory in in the WWW path. create a directory called classes and place the files Inline.class, DataNode.class and WriteCanvas.class in that directory.

Now the URL of our Database Browser is http://hostname/[directory of index.html]/

Chapter 6

Conclusions

6.1 Summary

In this thesis we have developed a tool, Database Browser, which uses some most commonly used hierarchical metaphors such as Tabbed indices, Inline Expansions and Tree structures to browse databases. The idea behind providing multiple metaphors is, to enable the user to choose a metaphor with which he is comfortable. In the development of this tool we have taken care of two types of considerations: considerations related to networking and considerations related to browsing and visualization.

The following considerrations are taken care of in networking part:

- Minimizing the load on the server.
- Minimizing the data to be sent to the client.
- Utilizing the resources on the client as much as possible.

The goals: minimizing the load on the server and tapping the resources on the client machine, are achieved by leaving all the visualization issues to client-side scripts. Server-side scripts just need to send the data required by client-side scripts. The goal: minimizing the the data to be sent on the network, is achieved by using multilevel indices to browse a table.

The following considerations are taken care of in browsing part:

- Using very simple and easy to use metaphors to present data.
- Selection of the most commonly used techniques so that the user need not waste time on learning how to use the technique.
- Selection of the most generic metaphors so that they can be used to present any data.

Above goals are achieved by selecting the metaphors such as Inline Expansions, Tree Structures and Tabbed indices. These metaphors are the most widely used, the easiest to use and the most generic.

1.2 Further extensions

The following are the possible extensions to this thesis.

- Our tool can be used to browse one table at a time. It cannot be used for complex queries involving multiple tables. This tool can be extended to incorporate complex queries.
- Data visualization techniques, such as Bar charts, Pie charts, Line charts, planar glyphs, 2-D and 3-D isosurfaces can be implemented.
- The server-side scripts that interact with the database, can be implemented in Perl-DBI(DataBase Interface) so that the scripts can work with any type of database without any need for modification.
- Currently the server-side scripts build the indices on the necessary tables. If the database indices are directly accessible this would not be necessary and the service responces can be considerably speeded up.

Bibliography

[1] Oracle WebServer Users Guide. Oracle Corporation, 1995.

[2] Oracle7 Server Administrators Guide. Oracle Corporation, 1995.

[3] Oracle7 Server Concepts. Oracle Corporation, 1995.

[4] Oracle7 Server SQL Reference. Oracle Corporation, 1995.

[5] PL/SQL User's Guide and Reference. Oracle Corporation, 1995.

[6] BOUTELL, T. CGI Programming in C and Perl. Addission-Wesley Developers Press, 1996.

[7] BOUTELL, T. World Wide Web FAQ. http://www.boutell.com/faq, 1996.

[8] HUGHES, K. World Wide Web Guide. http://www.eit.com/web/www.guide/, 1994.

[9] MOHNKERN, K. Beyond the interface metaphor. SIGCHI Bulletin 29, 2 (1997), 11–15.

[10] MURTHY, P. G. Information services through a web server. Tech. Rep. MT-CS-97-23, IIT, Kanpur, 1997.

[11] NAUGHTON, P., AND SCHILDH, H. The Complete Java Reference. Osborne McGraw-Hill, 1997.

[12] ROWE, AND EFF. Accessing a Database server via the World Wide Web. http://www.cscsun1.larc.nasa.gov/beowulf/db/Web_access.html, 1996.

[13] SHIRAN, Y., AND SHIRAN, T. Learn Advanced JAVASCRIPT Programming. bpb publications, 1998.

[14] TIM BERNERS-LEE, E. A. HyperText Transfer Protocol - HTTP/1.1, RFC 2068. MIT, Laboratory for Computer Science, 1997.